

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for arbitrating between a plurality of access requests issued in relation to a resource by a plurality of requestors, wherein each request can be one of at least two types, a first of the types having a higher latency associated with its performance than at least some of the other types, the method including the steps of:
 - (a) receiving a plurality of the access requests ~~on a plurality of data buses, the first type of access request being received on a different data bus than the other types of access requests;~~
 - (b) maintaining a current pointer that points to a current timeslot in a timeslot list, and at least one lookahead pointer that points to a future timeslot in the timeslot list; and
 - (c) in the event an access request as arbitrated via the lookahead pointer is of the first type, initiating performance of the access request earlier than the position in the list suggests it would be performed should it be started when the current pointer reached the timeslot, wherein, in step (c), the earlier position is selected so as to not be adjacent a position in the list for performance of another access request of the first type.
2. (Original) A method according to claim 1, wherein step (c) includes the substep of performing the access request indicated by the lookahead pointer immediately after the access request indicated by the current pointer is performed.
3. (Original) A method according to claim 1, wherein the first type of access request is a memory write request.
4. (Previously Presented) A method according to claim 3, wherein step (c) includes the substep of performing the access request indicated by the lookahead pointer immediately after the access request indicated by the current pointer is performed.
5. (Original) A method according to claim 1, wherein the number of timeslots between the timeslot indicated by the lookahead pointer and the timeslot indicated by the current pointer takes into account a latency difference between performing an access request of the first type and at least one of the other access request types.

6. (Currently Amended) An integrated circuit including:
- a plurality of operative units, each of which is capable of issuing a request for access to a memory accessible by the integrated circuit; and
- an timeslot arbitrator for arbitrating between requests issued by the operative units for access to the memory, wherein each request can be one of at least two types, a first of the types having a higher latency associated with its performance than at least some of the other types, the timeslot arbitrator being configured to:
- (a) ~~receive a plurality of the access requests on a plurality of data busses, the first type of access request being received on a different data bus than the other types of access requests;~~
 - (b) maintain a current pointer that points to a current timeslot in a timeslot list, and at least one lookahead pointer that points to a future timeslot in the timeslot list; and
 - (c) in the event the access request as arbitrated via the lookahead pointer is of the first type, performing the access request earlier than the position in the list suggests it should be performed should it be started when the current pointer reached the timeslot,
wherein, in step (c), the earlier position is selected so as to not be adjacent a position in the list for performance of another access request of the first type.
7. (Original) An integrated circuit according to claim 6, wherein the first type of access request is a memory write request.
8. (Cancelled)
9. (Original) An integrated circuit according to claim 6, wherein the number of timeslots between the timeslot indicated by the lookahead pointer and the timeslot indicated by the current pointer takes into account a latency difference between performing an access request of the first type and at least one of access request types.
- 10-18. (Cancelled)